ENSEIRB 2006-2007 - Département informatique - Première année
IF114 - Automates finis et applications

Examen (session1)

Veillez à soigner la rédaction de vos solutions, particulièrement vos justifications et vos preuves. Soyez précis et concis. Bon courage et bonne chance !

Exercice 1 (4 points)

On rappelle ci-dessous la **négation** du lemme de l'étoile.

> Si pour tout $N \in \mathbb{N}$, il existe $w \in L$ vérifiant :
> 1. $|w| > N$,
> 2. et pour **toute décomposition** $w = x.u.y$ avec $|u| > 0$ et $|x.u| \leq N$, il existe $n \in \mathbb{N}$ tel que $x.u^n.y \notin L$
>
> alors, L n'est pas un langage régulier.

Pour chacun des langages suivants, prouvez qu'il est régulier en donnant l'automate fini **minimal** qui l'accepte, ou qu'il est irrégulier en utilisant la négation du lemme de l'étoile.

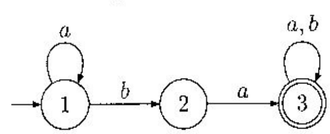
1. $L_1 = \{(ab)^k \mid k \in \mathbb{N}\}$
2. $L_2 = \{a^k b^m \mid k, m \in \mathbb{N} \text{ et } k \leq m\}$
3. $L_3 = \{a^k b^m \mid k, m \in \mathbb{N}\}$

Exercice 2 (4 points)

Pour tout automate fini déterministe $A = (Q, q_0, \Sigma, \delta, F)$, nous définissons $\overline{A} = (\overline{Q}, q_0, \Sigma, \overline{\delta}, \overline{F})$ par :

- $\overline{Q} = Q \cup \{\chi\}$ où $\chi \notin Q$ est un nouvel état
- pour tout $q \in \overline{Q}$ et $s \in \Sigma$, si $\delta(q, s) = q'$, alors $\overline{\delta}(q, s) = q'$, inversement, si $\delta(q, s)$ n'est pas définie, alors $\overline{\delta}(q, s) = \chi$
- enfin, $\overline{F} = \overline{Q} \setminus F$.

1. Soit A l'automate fini ci-dessous. Représentez l'automate $\overline{A}$ correspondant.



2. Prouvez que, quel que soit A, $\overline{L(A)} = L(\overline{A})$: le langage de A est le complémentaire du langage de $\overline{A}$ (remarque : aucune induction n'est nécessaire ici).

Exercice 3 (7 points)

On définit l'opérateur de *soustraction préfixe* sur les expressions régulières de la façon suivante :

- $\varepsilon^{-1}\beta = \beta$
- $a^{-1}\emptyset = \emptyset$
- $a^{-1}\varepsilon = \emptyset$
- $a^{-1}a = \varepsilon$
- $b^{-1}a = \emptyset$
- $a^{-1}(\alpha + \beta) = (a^{-1}\alpha) + (a^{-1}\beta)$
- $a^{-1}(\alpha\beta) = (a^{-1}\alpha)\beta$ si $\varepsilon \notin L(\alpha)$
- $a^{-1}(\alpha\beta) = (a^{-1}\alpha)\beta + (a^{-1}\beta)$ si $\varepsilon \in L(\alpha)$
- $a^{-1}(\alpha^*) = (a^{-1}\alpha)\alpha^*$

pour tous symboles $a, b \in \Sigma$ distincts et toutes expressions régulières α, β sur Σ.

L'opération $a^{-1}\alpha$ consiste à retirer a au début des mots définis par α. Par exemple :

$$\begin{aligned}
a^{-1}(\underbrace{a^*}_{\alpha}\underbrace{ba(a+b)^*}_{\beta}) &= (a^{-1}a^*)ba(a+b)^* + a^{-1}(ba(a+b)^*) \\
&= (a^{-1}aa^*)ba(a+b)^* + (a^{-1}b)a(a+b)^*) \\
&= (\varepsilon a^*)ba(a+b)^* + \emptyset a(a+b)^* \\
&= a^*ba(a+b)^* + \emptyset \\
&= a^*ba(a+b)^*
\end{aligned}$$

Notons que $\varepsilon\alpha = \alpha$ et $\emptyset + \alpha = \alpha$ et $\emptyset\alpha = \emptyset$ pour toute expression régulière α.

1. Pour remplir le tableau suivant, on débute avec l'expression régulière $a^*ba(a+b)^*$ (première ligne) à laquelle on applique successivement a^{-1} (première colonne) puis b^{-1} (deuxième colonne). Si une nouvelle expression régulière est construite par ce calcul, on crée une nouvelle ligne dans le tableau, puis on lui applique a^{-1} et b^{-1}, et ainsi de suite. Complétez le tableau ci-dessous (le nombre de lignes est exact).

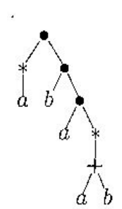
	a	b
$a^*ba(a+b)^*$	$a^*ba(a+b)^*$	

2. Représentez l'automate dont la relation de transition est définie par le tableau précédent. Son état initial est $a^*ba(a+b)^*$, et ses états finaux sont les expressions régulières α telles que $\varepsilon \in L(\alpha)$. Quelles propriétés remarquables possède cet automate ? Que représente l'expression régulière de chacun des états ?

3. Écrivez en langage naturel un algorithme qui calcule l'automate (c'est à dire le tableau) depuis une expression régulière α quelconque. On supposera connue une fonction qui calcule $a^{-1}\alpha$ pour a et α quelconques.

Exercice 4 (5 points)

Afin d'implanter l'opération de soustraction préfixe définie à l'exercice précédent, nous devons disposer d'une représentation des expressions régulières. Celles-ci sont généralement représentées par des arbres :



pour l'expression $a^*ba(a+b)^*$

Nous supposons donc l'existence d'un type `tree` et de la fonction :

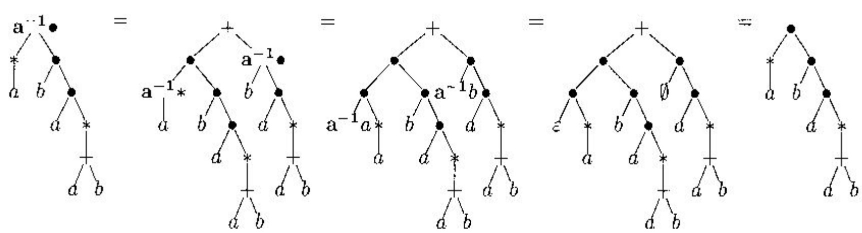
```
#define EPSILON   0
#define EMPTY     1
#define CONCAT    '.'
#define UNION     '+'
#define KLEENE    '*'

tree *node(char op, tree *l, tree *r);
```

L'arbre précédent se construit donc comme suit :

```
node(CONCAT,
     node(KLEENE,
          node('a',NULL,NULL),
          NULL),
     node(CONCAT,
          node('b',NULL,NULL),
          node(CONCAT,
               node('a',NULL,NULL),
               node(KLEENE,
                    node(UNION,
                         node('a',NULL,NULL),
                         node('b',NULL,NULL)),
                    NULL))))
```

Le calcul de a^{-1} peut alors être effectué sur ces arbres, comme montré ci-dessous :



On dispose des fonctions :

```
char label(tree *t);
tree *left(tree *t);
tree *right(tree *t);
```

qui retournent respectivement l'étiquette, le fils gauche et le fils droit d'un nœud de l'arbre.

1. Écrivez en langage C la fonction `tree *prefix_sub(char c, tree *t);` qui calcule $c^{-1}t$ **sans réaliser les simplifications** par ε et $\emptyset$ correspondant à la dernière étape ci-dessus. S'il vous est nécessaire de définir de nouvelles fonctions, donnez leurs signatures et les traitements réalisés, sans en écrire le code.

2. Indiquez comment modifier votre fonction `prefix_sub` pour y ajouter les simplifications $\emptyset\alpha = \emptyset$ et $\emptyset + \alpha = \alpha$ et $\varepsilon\alpha = \alpha$